

[INICIAR SESIÓN](#)[NUESTROS PLANES](#)[TODOS LOS  
CURSOS](#)[FORMACIONES](#)[CURSOS](#)[PARA  
EMPRESAS](#)[ARTÍCULOS DE TECNOLOGÍA > FRONT END](#)

# CSS: animaciones en Transition y Animation



Mario Souto

27/10/2020

Hoy en día las animaciones en los sitios web van mucho más allá de los simples detalles, algunas marcas tienen su marca registrada en las animaciones, al igual que el propio Google después de su último rebrand.



Abordaremos técnicas sobre cómo hacer animaciones a través de CSS y al final dejaré un dato sobre cómo puedes tener ideas para llevar animaciones como la del logotipo de Google a una página web.

## Transition: La base de las animaciones basadas en interacción

En la web, se pueden animar muchas cosas, básicamente todo lo que posee **etapas**. Ya sea un formulario de registro cuando cambiamos entre fases, un producto que se agrega a un carro de campos, la aparición de partes de la página según un scroll. En la práctica, la mayoría de estas animaciones ocurre después de que el usuario realiza una acción.

Con eso en mente, exploremos la propiedad `transition` que si se usa junto con los pseudo selectores `:hover` y `:focus` del CSS.

En la práctica el `transition` se puede aplicar a todos los valores que cambian entre los estilos estándares de un elemento y alguna variación aplicada, ya sea a través de alguna

acción de pseudo selector (:hover y :focus) o por algún atributo o clase que se agrega a una tag y exista un selector CSS aplicando alguna variación de estilo.

*Solo se pueden animar valores cuantificables. Esto significa que no es posible hacer una animación de display pasando de none a block, sino de opacity de 0 a 1;*

Centrándonos en la idea de que las Animaciones son intercambios de estados donde podemos visualizar cómo algo pasó del punto A a un punto B, animar el background es solo una posibilidad, podemos manipular el ancho y alto de los elementos, su posición en la página y una propiedad que coincide súper bien con el transition es el transform.

Como el [transform](#) puede aplicar diferentes comportamientos a un elemento como cambiar su escala, cambiar su posición en la página o incluso aplicar una distorsión con skew() esto abre toda una gama de posibilidades.

## Formas comunes de utilizar el transition:

- `transition: 1s;`: aplica una animación de 1 segundo a TODAS las propiedades de un selector, es importante especificar qué propiedades se animarán para evitar algunos problemas de desempeño o propiedades desnecesarias al consumir recursos en animaciones (que normalmente pueden exigir mucho de computadoras de usuarios con hardware menos potentes).
- `transition: transform 1s, opacity 0.3s;`: Aplica una animación de 1 segundo para la propiedad transform, y una de 0,3 segundos para el opacity. Todas las demás propiedades no se ven afectadas.
- También es posible especificar un delay para el transition con `transition-delay`, además de especificar la duración y las propiedades a través de propiedades independientes, a diario terminamos usando la forma de los ejemplos anteriores, pero, [recomiendo que veas en MDN cómo trabajar con estas variaciones](#)

## Animation: Animando automáticamente y con más pasos

Como vimos anteriormente, para animar algo con transition, dependemos de alguna acción del usuario; ahora, ¿y si queremos que la animación ocurra sola de forma continua? Tenga en cuenta que eso es muy similar a algunos ítems que existen para atraer la atención, como flechas que indican que se puede pulsar en algún punto en algunos sitios. Para que esto sea posible estamos utilizando la propiedad animation.

Para entender cómo hacer una animación así, primero debemos comprender la idea de trabajar con keyframes. Básicamente definiremos que dado un intervalo de tiempo entre el inicio y el final del mismo se aplicarán determinadas propiedades CSS.

```
@keyframes go-back {  
  0% {  
    transform: translateX(100px);  
  }  
  100% {  
    transform: translateX(0);  
  }  
}
```

Y para asociar estos keyframes con un elemento HTML, necesitamos agregar esta animación go-back, o cualquier otra, por medio de la propiedad animation:

```
.cuadrado {  
  animation: go-back 1s;  
}
```

Pero, de esta forma la animación solo ocurre una vez, si estamos interesados en que suceda infinitamente podemos agregar la propiedad animation-iteration-count: infinite; en lugar de infinite, podemos pasar un número máximo de veces que nuestra animación se repetirá, y si quisieras hacer que la animación vaya hacia adelante y hacia atrás desde el punto final hasta el principio y viceversa, simplemente agrega la propiedad animation-direction: alternate;.

## Animaciones comunes en sitios web:

### Bounce

```
.ball {  
  margin-top: 50px;  
  border-radius: 50%;  
  width: 50px;  
  height: 50px;
```

```
background-color: cornflowerblue;
border: 2px solid #999;
animation: bounce 1s infinite alternate;
-webkit-animation: bounce 1s infinite alternate;
}
@keyframes bounce {
  from {
    transform: translateY(0px);
  }
  to {
    transform: translateY(-15px);
  }
}
@-webkit-keyframes bounce {
  from {
    transform: translateY(0px);
  }
  to {
    transform: translateY(-15px);
  }
}
```

## FadeIn:

```
.ball {
  margin-top: 50px;
  border-radius: 50%;
  width: 50px;
  height: 50px;
}

@keyframes fadein {
  from { opacity: 0; }
```

```
to { opacity: 1; }  
}
```

## Interceptar animations y transitions por vía JavaScript

Como vimos anteriormente, es posible tener varios niveles de control en animaciones con CSS, sin embargo, en caso de que desee tener un control más preciso a través de JavaScript por medio de un `element.addEventListener` es posible monitorear cuando termina una transition o animation:

- [https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/animationend\\_event](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/animationend_event)  
[https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/transitionend\\_event](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/transitionend_event)

## Mejorando el desempeño de animaciones con CSS

Esta publicación es más introductoria, pero todavía creo que vale la pena mencionar que trabajar con el desempeño en animaciones puede requerir un poco de conocimiento de cómo funciona el navegador, y un poco de cómo funciona el CSS en navegadores.

Ya te adelanto que hay una propiedad que puede obrar milagros en algunos casos para mejorar el desempeño de las animaciones, pero que sólo debería usarse en último caso:

- [will-change](#)

*Si encuentra alguna animación que realmente obstaculice la experiencia del usuarios en devices menos potentes, puede deshabilitarla utilizando [@prefers-reduce-motion](#).*

## Libs adicionales

- [animate.css](#): Esta fue probablemente la lib que más me ayudó en la época de agencia y lo mejor es que hay ports de ella para trabajar con los frameworks de moda a través de componentes y es una forma realmente genial de animar algo muy rápidamente o de abrir el código fuente y aprender [cómo se hacen algunas animaciones como por ejemplo, el bounceIn](#)
- [Tweenmax - Greensock](#): Si quieres hacer algo que tenga un control más fino de lo que se anima en cada momento o según el cambio X del usuario, la mejor alternativa es acudir a una lib que te ayude a preparar estas "escenas". Tweenmax es una gran

opción, solo dejo un punto de observación porque tiene algunas restricciones de uso gratuito.

- [Lottie del Airbnb](#): Esta es la lib del momento. Use el plugin para premiere y exportar para su uso en la web, pero podemos abordar eso en otro artículo.

Si te interesa este tema, aquí en [Alura](#) tenemos capacitaciones de front-end, para principiantes y para aquellos que ya son desarrolladores web. En ellas, verás cómo programar en Javascript, HTML y CSS para construir sitios web.

ARTÍCULOS DE TECNOLOGÍA > FRONT END

## En Alura encontrarás variados cursos sobre Front End. ¡Comienza ahora!

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas



Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90



218 cursos



Videos y actividades 100% en Español



Certificado de participación



Estudia las 24 horas, los 7 días de la semana



Foro y comunidad exclusiva para resolver tus dudas



Acceso a todo el contenido de la plataforma por 12 meses



**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A

CNPJ 05.555.382/0001-33

## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento